WHAT IS CLAIMED IS:

1. A processing core, comprising:

a first source register including a plurality of first operands;

a plurality of second operands;

a bitwise inverter coupled to at least one of the first plurality of operands and the second plurality of operands;

a destination register including a plurality of results;

a plurality of arithmetic processors respectively coupled to the first operands, second operands and results, wherein each arithmetic processor computes one of a sum and a difference of the first operand and a respective second operand.

2. The processing core of claim 1, further comprising an integrated circuit which includes the first source register, destination register and arithmetic processor.

3. The processing core of claim 1, wherein the plurality of second operands are equal to each other.

4. The processing core of claim 1, wherein each arithmetic processor computes at least one of:

the result of the first operand plus the respective second operand plus a positive integer; and

the result of the first operand minus the respective second operand minus a positive integer.

5. The processing core of claim 1, wherein the plurality of second operands includes a signed immediate value.

6. The processing core of claim 1, further comprising a prescaler which scales each of the plurality of second operands.

7. The processing core of claim 1, wherein a first width of the first source register is a positive integer multiple of a second width of the first operand.

8. The processing core of claim 1, wherein the sum and the difference are performed on the same carry look-ahead adder.

9. A method for performing arithmetic processing, the method comprising the steps of:

loading a first and second operands from a primary source register;

loading a third and fourth operands from a secondary source register;

scaling the third and fourth operands according to a predetermined scaling factor;

performing an arithmetic function on the first and third operands to produce a first result;

performing the arithmetic function on the second and fourth operands to produce a second result; and

storing the first and second results in a destination register.

10. The method for performing arithmetic processing of claim 9, further comprising a step of inverting the third and fourth operands.

11. The method for performing arithmetic processing of claim 9, further comprising a step of adjusting at least one of the first and second results to avoid saturation of the destination register.

12. The method for performing arithmetic processing of claim 9, wherein the step of performing an arithmetic function on the first and third operands comprises calculating the first operand plus the second operand plus a positive integer.

13. The method for performing arithmetic processing of claim 9, wherein the step of performing an arithmetic function on the second and fourth operands comprises calculating the second operand minus the fourth operand minus a positive integer.

14. The method for performing arithmetic processing of claim 9, wherein the third and fourth operands are the same immediate value.

15. The method for performing arithmetic processing of claim 9, wherein the predetermined scaling factor is divisible by two.

16. The method for performing arithmetic processing of claim 9, wherein the two performing steps are performed, at least partially, coextensive in time.

17. The method for performing arithmetic processing of claim 99, wherein the two performing steps use a ripple look-ahead adder.

18. A method for performing arithmetic processing, comprising the steps of:

loading a first and second operands from a primary source register;

loading an immediate value;

performing an arithmetic function on the first operand and immediate value to produce a first result;

performing the arithmetic function on the second operand and immediate value to produce a second result; and

storing the first and second results in a destination register.

19. The method for performing arithmetic processing of claim 18, wherein the immediate value has a width of nine bits.

20. The method for performing arithmetic processing of claim 18, wherein the immediate value has a width of thirteen bits.

21. The method for performing arithmetic processing of claim 18, wherein the two performing steps are performed, at least partially, coextensive in time.

22. The method for performing arithmetic processing of claim 18, further comprising a step of adjusting at least one of the first and second results to avoid saturation of the destination register.